REMARKS

In a final office action dated November 30, 2005, the Examiner rejected claims 1-5 and 7-26 under 35 U.S.C. 102(b) as anticipated by Shrader et al.(U.S. Patent 5,870,611).

In order to simplify the issues remaining herein, applicants have cancelled independent claims 1, 16, 17 and 23, and the claims dependent thereon.

Independent claims 12 and 18 has been amended to include the limitations previously contained in dependent claims 15 and 19, respectively, i.e., that the instruction set contained in the upgrade object or installation object is a script which is compiled and executed by the script processor or instruction processing module. Claims 15 and 19 have been cancelled as superfluous. As amended, claims 12 and 18 are allowable.

A brief background discussion will be helpful in appreciating the contribution made by applicants' invention. Maintaining software in numerous and diverse customer installations is a major problem for software vendors and their customers. Electronic distribution of software upgrades, via the Internet or other methods, is well known. But distribution is only one step in upgrading a computer program. The upgrade must further be installed on the user's system. Each user's system may be different, employing different hardware configurations, different operating systems, different software applications, and so forth. Each of multiple software applications or operating systems may further be at different levels and versions. Software can be installed manually on multiple systems by trained personnel; this technique is still used, particularly for large systems or installations which are relatively small in number, but it is obvious that this technique becomes very expensive and burdensome as the number of installations increases.

Docket No.:   ROC920010099US1
Serial No.:   09/821,920

For mass produced software, it is common to provide automated installation programs, which are distributed to the users with the new applications or upgrades to existing applications, and automatically install the new applications or upgrades. Generally, a user simply invokes the installation software, and the installation software does the rest. In some cases, the installation software may request installation parameters or other information from the user, but generally the process is automated to avoid requiring special knowledge or expertise of the user.

Conventional installation programs are just that, i.e., they are *executable programs* which perform an installation function automatically. Because such conventional installation programs must be designed to accommodate a wide variety of system configurations and to execute with a minimum of end user input, these are complex programs. Development of such complex conventional installation programs is a very involved process, like developing any complex application software. Furthermore, the installation code itself occupies considerable memory volume, and transmission of such code by electronic means consumes considerable bandwidth of the transmission medium. When multiplied by the numerous upgrades and the number of systems to be upgraded, this consumption of transmission bandwidth can become a significant problem.

Applicants' invention involves an improved upgrade process, whereby upgrade objects in the form of scripts are distributed in place of executable programs. I.e., the "upgrade objects" or "installation objects" of applicant's invention are not executable programs, but are scripts which require the use of a separate script processor. The script processor is distributed in advance, preferably as part of the computer program, but only need be distributed once. Thus, the script processor is not required to be distributed with each upgrade, effectively reducing the transmission bandwidth required to support upgrades. The script processor is executable code which reads the script contained in the update object and performs the upgrade according to the

Docket No.: ROC920010099US1
Serial No.: 09/821,920

script. I.e., the script processor is essentially a compiler or interpreter which interprets the script commands and performs the functions specified therein.

Shrader discloses a technique for building an "installation plan object" for installing software, using object-oriented programming techniques. Essentially, *Shrader* starts with an empty container object and adds various "child objects" from a library of such objects in order to customize the installation.

It should be understood that **Shrader is describing a process that takes place at the software developer**. I.e., the software developer, in order to develop an installation program, builds the program from *Shrader*'s container object and child objects, using object-oriented programming concepts. However, the installation program, once built, is **executable code**. This code is then distributed in the conventional fashion, and executed on the individual customer systems to install the software.

Shrader does not anticipate applicants' claims because, inter alia, *Shrader* does not teach the use of a script processor, separate from the upgrade object, which translates script in the upgrade object to execute the upgrade or installation.

Applicants' representative claim 12, as amended, recites:

12.   (Currently Amended)   A method of upgrading a computer program on a computer system, the computer program including a script processor, the method comprising:
        creating an upgrade object associated with the computer program, the upgrade object including an *instruction set adapted for use by the script processor* to upgrade the computer program; and
        transmitting the upgrade object to the computer system; and
        instructing an end user to *execute the instruction set with the script processor*;
        wherein the *instruction set comprises a script* and wherein the *script processor is adapted to compile and execute the script*. [emphasis added]

Docket No.:   ROC920010099US1
Serial No.:   09/821,920

Independent claim 18 contains analogous limitations to the italicized limitations above.

The Examiner appears to be reading various parts of the claims in isolation on certain elements of *Shrader*, without considering the interaction of the various elements. The Examiner calls *Shrader*'s "proposed plan object" an "upgrade object", as recited in applicants' claim 12. Very well. Claim 12 further recites that the upgrade object includes "an instruction set adapted for use by the script processor". *Shrader* does indeed mention instructions in the upgrade object, but these appear to be executable instructions; there is no mention whatsoever of instructions "adapted for use by [a] script processor", as claimed by applicants.

The Examiner apparently considers the operating system the equivalent of a script processor. Applicants beg to differ. An operating system does not convert script in a script object into executable instructions. An operating system does not even convert executable instructions in an executable code file. It provides certain low-level functions which allocate system resources to processes, the processes executing the executable code. Applicants therefore submit that an operating system does not satisfy the limitations of a script processor; it does not "execute the instruction set with the script processor".

But even if there could be any doubt about the above, it can not possibly be said that *Shrader*'s operating system "compiles and executes" the script, as recited in amended claim 12. The Examiner appears to find this limitation in the mention of "Install Scripts" in *Shrader*. But here is where the Examiner's read stretches to the breaking point. The Examiner seems to have forgotten that the script processor has been read on the operating system. *Shrader*'s Install Scripts

Docket No.:   ROC920010099US1
Serial No.:   09/821,920

appear to be file designations for locating installation files.[1] There is no disclosure or suggestion that these are "compiled and executed" by the operating system.

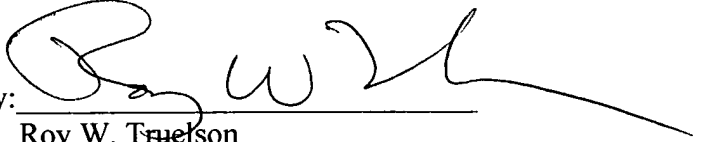For all of these reasons, claims 12 and 18, as amended, are not anticipated by *Shrader*.

Similarly, the claims are not obvious over *Shrader*. There is simply no equivalence between an operating system, which is well understood in the art, and applicants' claimed script processor (or instruction processing module) which **compiles and executes scripts**. Nothing in *Shrader* suggests the use of such a module to compile and execute scripted upgrade objects, thus reducing the volume of transmission bandwidth consumed during successive upgrades of a software application.

In view of the foregoing, applicants submit that the claims are now in condition for allowance and respectfully request reconsideration and allowance of all claims. In addition, the

---

[1] See *Shrader*, col. 9, lines 45-50. *Shrader* does not explain Install Scripts in any detail and the meaning of this term is obscure, but in any case there is no teaching or suggestion that an operating system (or indeed, anything else) "compiles and executes" the Install Scripts, as recited in applicants' claims.

Docket No.:   ROC920010099US1
Serial No.:   09/821,920

PATENT – AMENDMENT AFTER FINAL
Response Under 37 C.F.R. 1.116 – Expedited
Procedure – Examining Group 2192

Examiner is encouraged to contact applicants' attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

SAMUEL D. DULL III, et al.

By: _____

Roy W. Truelson
Registration No. 34,265

Telephone: (507) 202-8725

Docket No.:   ROC920010099US1
Serial No.:   09/821,920